RESOURCE ALLOCATION IN INTERNET OF THINGS CLOUD ENVIRONMENT USING SWARM OPTIMIZATION

A

Synopsis

Submitted in the partial fulfilment for the requirement of

Doctor of Philosophy

in

Computer Science

Submitted by

Amarjeet Singh Chauhan

Under the Supervision of

Dr. Sanjay Saini

Department of Physics and Computer Science
Faculty of Science

DEPARTMENT OF PHYSICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
DAYALBAGH EDUCATIONAL INSTITUTE
DAYALBAGH AGRA -282005

2022

Contents

1.0 Introduction 3
2.0 Optimization techniques for Resource Allocation 5
2.1 Genetic Algorithm 5
2.2 Particle Swarm Optimization 6
2.3 Ant Colony Optimization 7
2.4 Artificial Bees Colony Optimization 8
3.0 Internet of Things Cloud Environment 8
3.1 Cloud Computing 9
3.2 Fog Computing 9
4.0 Literature Review 9
5.0 Motivation 13
6.0 Proposed Research Objectives 14
7.0 Timeline 14
8.0 Platform 15
9.0 References 15

1.0 Introduction

Resource Allocation on the Internet of Things Cloud has become increasingly complex during recent years. A wide range of organizations and companies that deal with cloud environments is facing this problem. This is because constraints and objectives encountered in practice are highly variable and vary for each setting. Since the resource allocation problem in IoT Cloud Environment (1999) was introduced, performance has been the main constraint to focus on. But the problem is now more complex due to many other factors suchas task execution time, task transferring, task execution cost [1], etc.

Task scheduling (TS) is one of the most famous combinatorial NP-complete tasks. The efficiency of TS directly affects system performance. Various researchers are proposing various algorithms for the efficient allocation and scheduling of resources in the cloud. This has been found in past comparative studies of various evolutionary and swarm-based methods to perform task scheduling of resources such as ant colony, bee colony, genetic algorithms, particle swarm, and simulated annealing. Various modified scheduling algorithms such as improved multi-objective particle swarm optimization, modified ant colony optimization, and genetic algorithms were also analyzed [2].

Internet of Things is the integration of many technologies and sometimes called the Internet of Everything (IoE), Web-based components that collect, transfer, and the data they use from their ecosystem implanted sensors, processors, and communication hardware. IoT is projected to be the next big revolution Communication world. IoT aims to create a flawless network of billions of wireless devices that can communicate over the Internet. IoT ecosystem has millions of components with enormous diversity ranging from small sensors to large, powerful data center nodes, the dynamic execution environment, unambiguous nature of the data generated by smart objects, all these made IoT ecosystem as an atypical ecosystem. Acquiring physical data and transforming it into valuable information and/or services include many processes, such processes need a resource from the IoT ecosystem. For instance, some applications are latency-sensitive, and some need complicated processing like data and time series analysis [3].

IoT components like sensors have limited computing and energy resources, and they are not capable

of storing large data and carrying out complicated tasks. So, there is a need for powerful components to carry out the transformation process required by IoT applications. Such components may be smartphones, gateways, and data centers. So, it is better to consider the IoT ecosystem with a cloud platform and some powerful devices like gateways, edge nodes. For example, some applications are latency-sensitive, and some require complex processing such as data and time series analysis [4]. IoT components like sensors have limited computing and Energy resources, and they are not capable of large storage to complete data and complex tasks. So, need Powerful components to complete the change process Required by IoT applications. May contain such components as Smartphones, Gateways, and Data Centers. So, it is better to Consider the IoT ecosystem with a cloud platform and more powerful tools such as gateways, edge nodes. Resources may vary from physical Resources to software resources. Physical resources are storage, processor, bandwidth, energy, and many more, software resources can be procedures to perform information fusion, virtualization function, etc. Identifying and analyzing such resources from the IoT ecosystem makes resource allocation a challenging task [4].

The IoT ecosystem consists of millions of heterogeneous devices that are connected through a network. Efficient resource management is required to increase the quality of services. Network diversity and diversity of IoT devices make resource allocation a challenging task, many efficient algorithms and techniques are being proposed and used to solve the resource allocation problem [5].

Currently, IoT-enabled technologies are widely used for increasing crop productivity, generating significant revenue, and efficient farming. The development of the IoT paradigm helps in precision farming. Agricultural loT systems perform crop health monitoring, water management, crop security, farming vehicle tracking, automatic seeding, and automatic pesticide spraying over the agricultural fields. In an IoT-based agricultural system, different sensors necessarily have to be deployed over agricultural fields, and the sensed data from these sensors need to be transmitted to a centralized entity such as a server, cloud, or fog devices. Further, these data have to be processed and analyzed to provide various agricultural services. Finally, a user should be able to access these services from handheld devices or computers.

Advantages of IoT in agriculture

1. Modern technological advancements and the rapid developments in IoT components have gradually increased agricultural productivity. Agricultural IoT enables the autonomous execution of different agricultural operations. The specific advantages of the agricultural IoT are as follows:
2. Automatic seeding: IoT-based agricultural systems are capable of autonomous seeding and planting over the agricultural fields. These systems significantly reduce manual effort, error probability, and delays in seeding and planting.
3. Efficient fertilizer and pesticide distribution: Agricultural IoT has been used to develop solutions that are capable of applying and controlling the amount of fertilizers and pesticides efficiently. These solutions are based on the analysis of crop health.
4. Water management: The excess distribution of water in the agricultural fields may affect the growth of crops. On the other hand, the availability of global water resources is finite. The constraint of limited and often scarce usable water resources is an influential driving factor for the judicious and efficient distribution of agricultural water resources. Using the various solutions available for agricultural IoT, water can be distributed efficiently, all the while, increasing field productivity and yields. The IoT-enabled agricultural systems are capable of monitoring the water level and moisture in the soil, and accordingly, distribute the water to the agricultural fields.
5. Real-time and remote monitoring: Unlike traditional agriculture, in IoT-based farming, a stakeholder can remotely monitor different agricultural parameters, such as crop and soil conditions, plant health, and weather conditions. Moreover, using a smart handheld device (e.g., cellphone), a farmer can actuate on-field farming machinery such as a water pump, valves, and other pieces of machinery.
6. Easy yield estimation: Agricultural IoT solutions can be used to record and aggregate data, which may be spatially or temporally diverse, over long periods. These records can be used to come up with various estimates related to farming and farm management. The most prominent among these estimates is crop yield, which is done based on established crop models and historical trends.
7. Production overview: The detailed analysis of crop production, market rates, and market demand are essential factors for a farmer to estimate optimized crop yields and decide upon the essential steps for future cropping practices. Unlike traditional practices, IoT-based agriculture acts as a

force multiplier for farmers by enabling them to have a stronger hold on their farming as well as crop management practices, and that too mostly autonomously. Agricultural IoT provides a detailed product overview on the farmers' handheld devices.

2.0 Optimization Techniques for Resource Allocation

2.1 Genetic Algorithm

A genetic algorithm is a technique of scheduling in which the tasks are assigned resources according to individual solutions (which are called schedules in the context of scheduling), which tells about which resource is to be assigned to which task. Several genetic algorithms have been developed to solve task scheduling problems. The main difference between most of these is the chromosomal representation used in the schedule. However, these existing algorithms are monolithic as they try to scan the entire solution space without considering how to minimize the complexity of the optimization.

The genetic algorithm (GA) was the first evolutionary method based on the laws of Mendelian inheritance and the principles of natural selection. GA offers several advantages over other methods for computationally intensive problems, provided appropriate adaptability functions and set operators are provided. GA defines a set of chromosomes (solutions) or sets of chromosomes called a population. This method then repeats the selection, mutation, and crossover operations until the stopping criteria are met. The result set is the solution set [6].

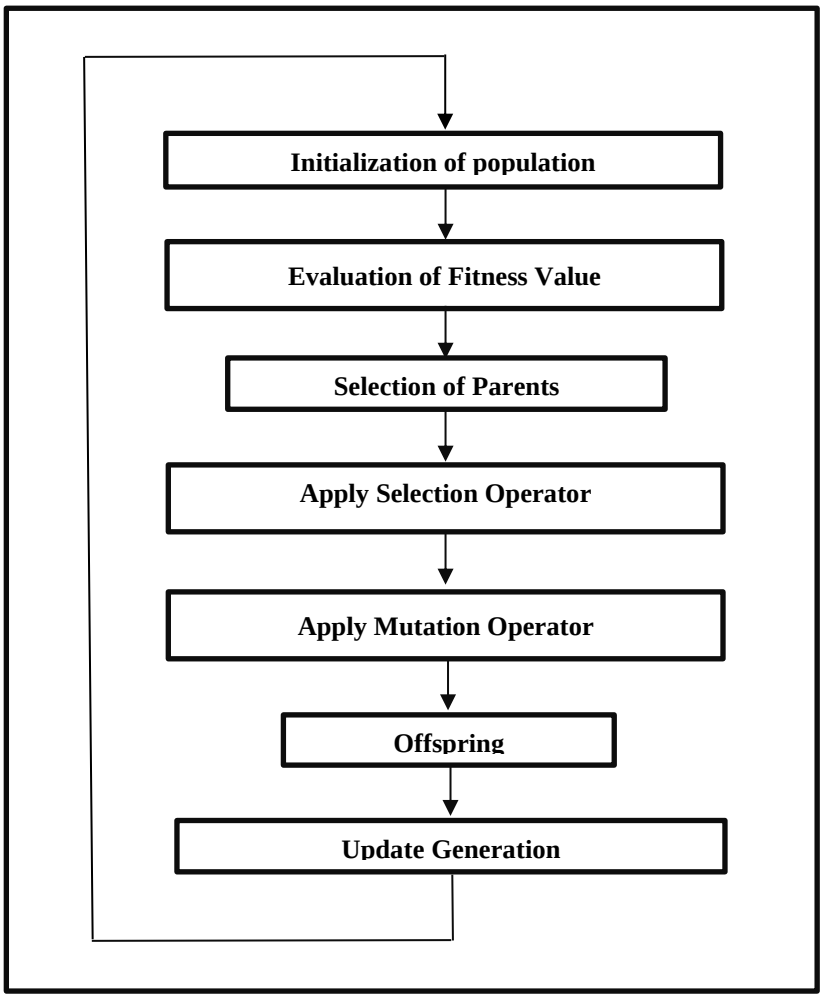

Figure 1: Steps of Genetic Algorithm

2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) has a heuristic approach that is applied to many large and complex problems such as power systems, data mining knowledge extraction, and task scheduling problems. PSO follows the principle of random search throughout the solution space using a large population depending on the problem area. Recently, PSOs provide better results than other well-known inefficient optimization solutions. Various methods applicable to task scheduling in this category include multipurpose task assignment using PSO, PSO-based heuristics for scheduling, PSO based on small position rules, and cost optimization using PSO.

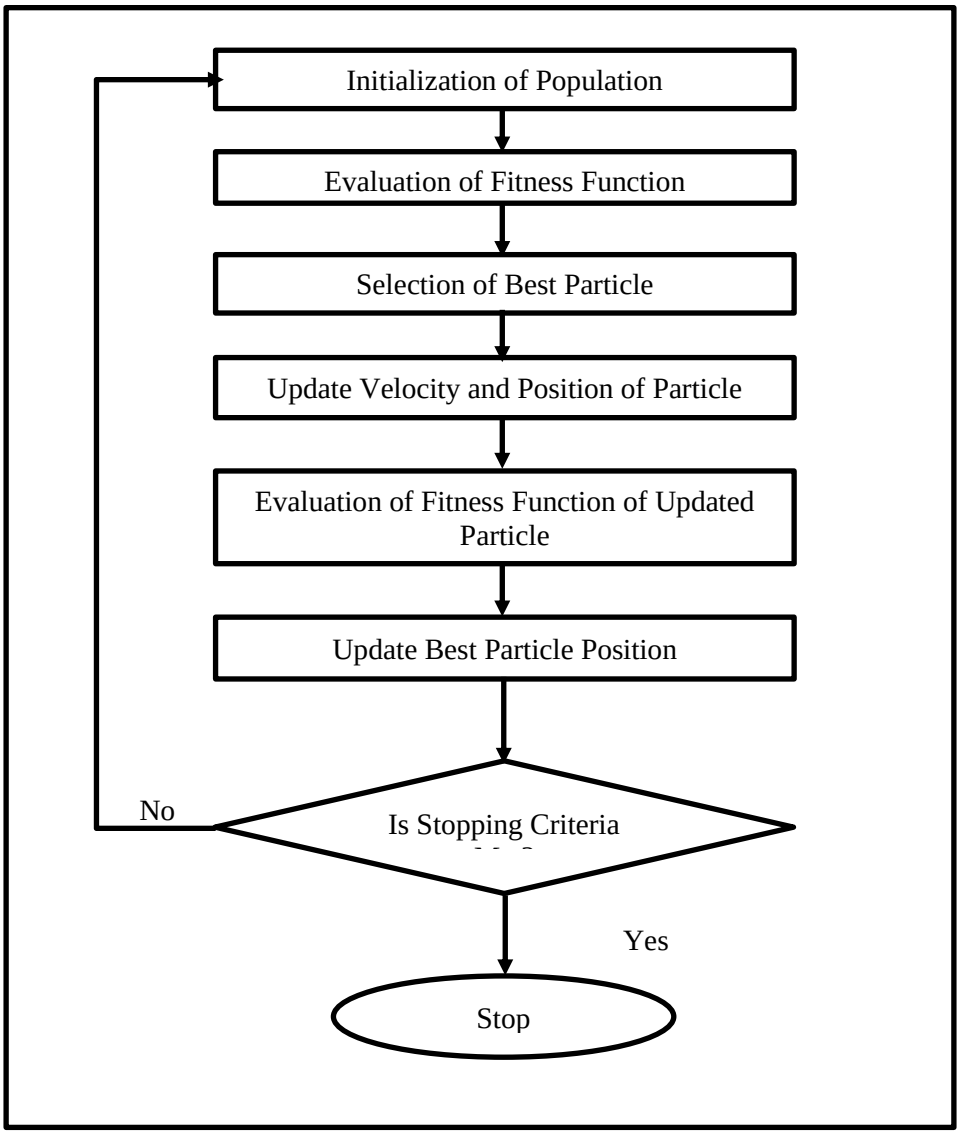

Figure 2: Steps of Particle Swarm Optimization

2.3 Ant Colony Optimization

Ant colony optimization (ACO) is a meta-heuristic technique inspired by the cooperative food search model of ants. A set of agents is used to implement the colony behavior of real ants. Ants use pheromone trails to cooperate and communicate. Ants use the build graph to iteratively solve problems. Here, the edges represent possible partial solutions that the ant can make according to the rules of stochastic state transitions. Selecting a partial or complete solution triggers a pheromone update rule. This rule provides a reaction mechanism that speeds convergence and prevents premature stagnation of the solution. Due to the detailed properties of ACO, various algorithms based on ACO metaheuristics have been applied to many complex optimization

problems [7]. The various methods used to schedule jobs in this category include ACO for load balancing and ACO for job matching.

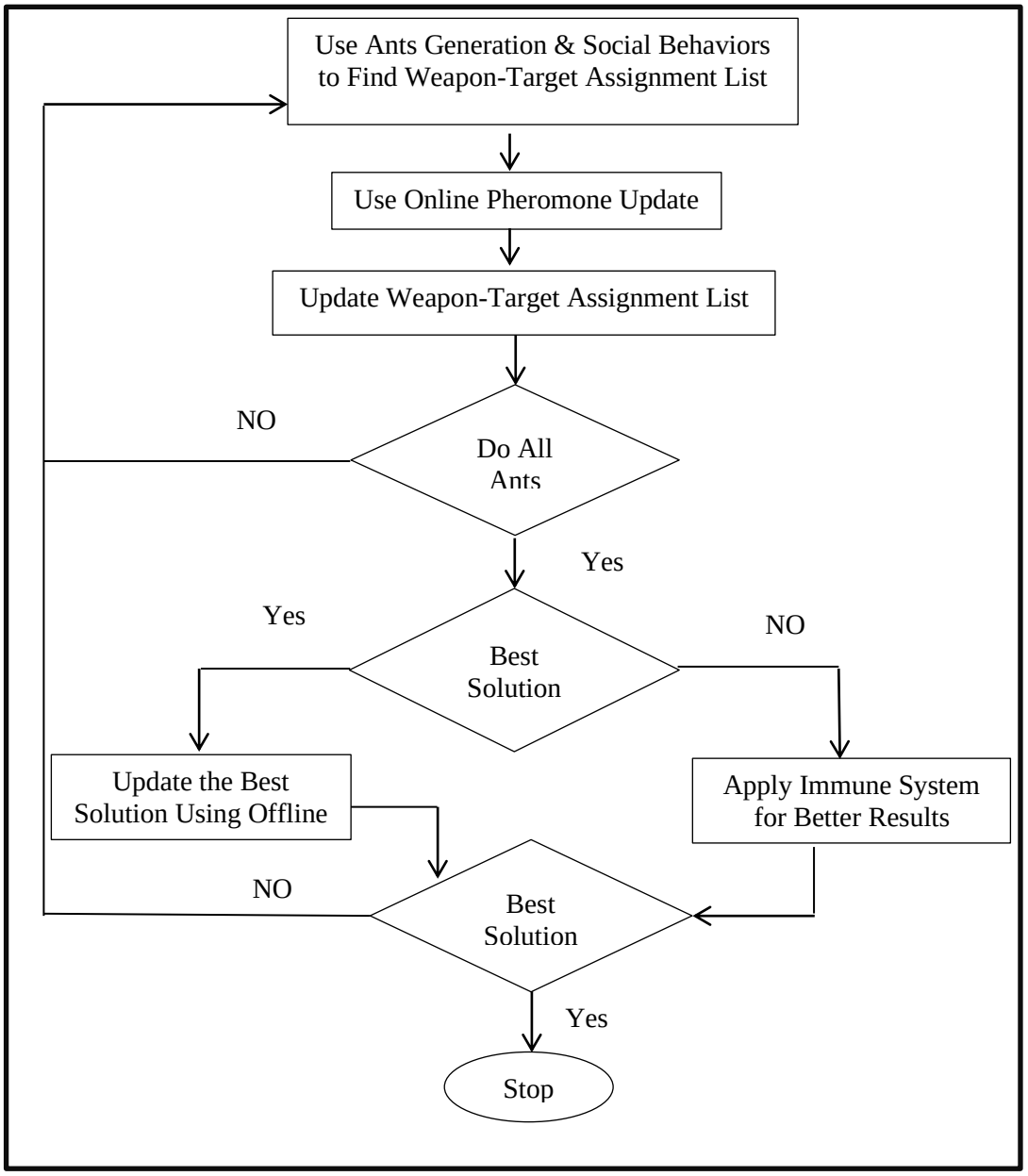

Figure 3: Steps of Ant Colony Optimizations

2.0 Artificial Bees Colony Optimization

The Artificial Bees Colony (ABC) is based on bee's activity when exchanging information with other bees in search of nectar. There are three types of agents: Employed Bees (EB), Reflective Bees (RB), and Scouts Bees (SB). EB live far from food sources and remember their surroundings. The audience receives this data from jobs and selects one of the food sources. Meanwhile, the scouts work to find a new honey source [8]. Various methods applied to job scheduling in this

category include modified bee lifespan algorithms and bee lifespan algorithms.

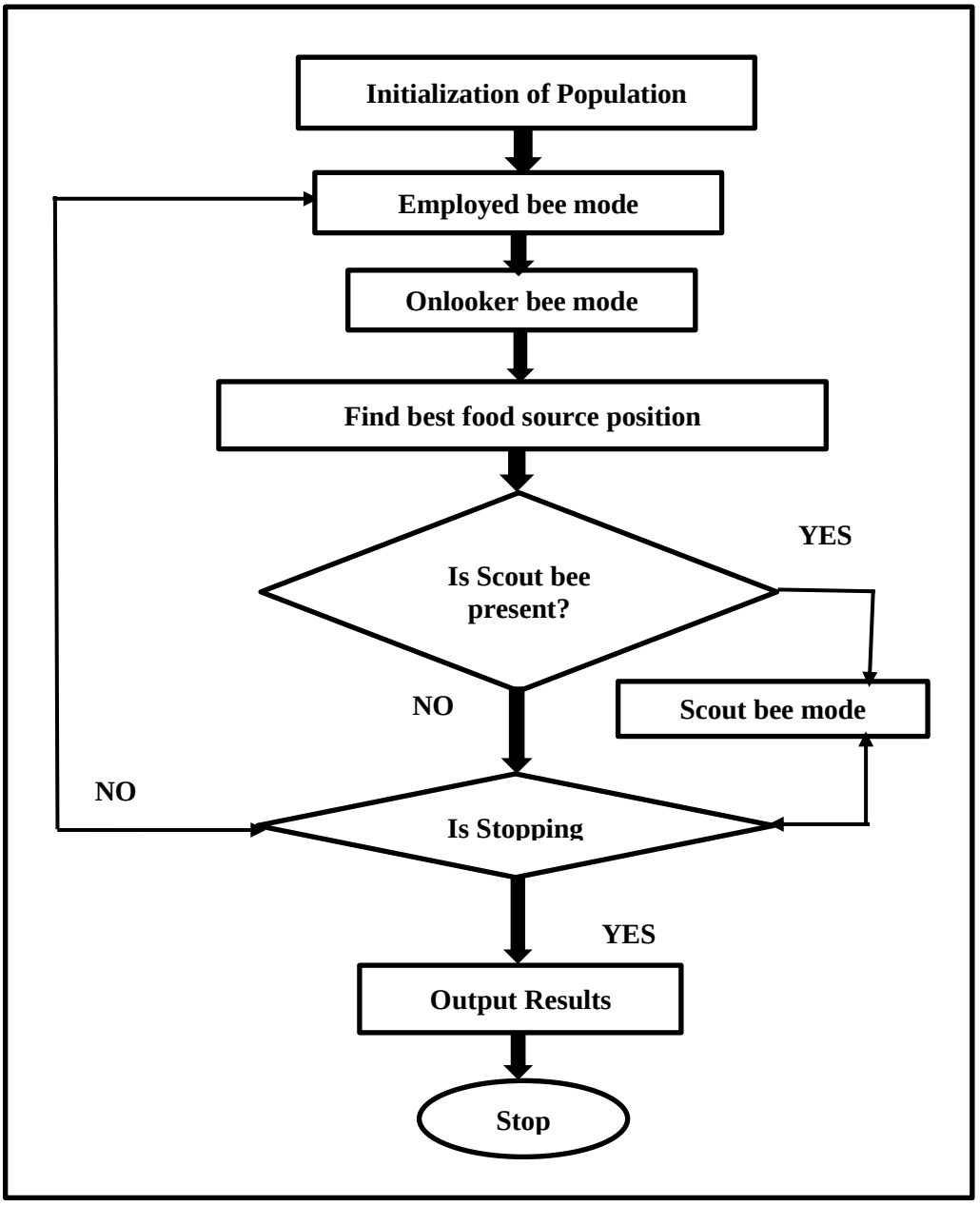

Figure 4: Steps of Artificial Bees Colony Optimization

3.0 Internet of Things Cloud Environment

Recently, the Internet of Things (IoT) has become one of the major revolutions in the information and communication technology industry. With IoT, an Internet connection to cater to many different applications and services such as traffic control, medical treatment, from traditional smart devices such as smartphones and tablets to devices and things (machines, sensors, vehicles, etc.) Spans a wide range. Energy Management, Health Care, Vehicle Network, etc. It produces a large

amount of data that needs to be processed, stored, and analyzed to obtain valuable information to meet the user's goals and needs. In addition, the number and scale of applications and services are growing rapidly, requiring processing capabilities that most powerful smart devices cannot meet at this time.

The cloud environment, known as a large resource hub, that allows users to share and flexibly provides resources, supports IoT development. Time-consuming and resource- intensive tasks can be minimized on powerful computing platforms such as cloud computing while leaving simple tasks to handle smart devices. However, when IoT and cloud computing is combined, new problems arise.

3.1 Cloud Computing

Cloud computing is the next-generation computing paradigm. It is a new computing technology that is increasingly integrated into the on-demand future of distributed computing [9] [10].

Cloud computing is becoming an important pillar for various Internet companies using virtualization principles. Several computing platforms have been proposed for big data storage and multiple parallel computing in cloud computing [11].

3.2 Fog Computing

Fog Computing [4], first proposed by Cisco, represents a new model of cloud computing, which can transform a network edge into a distributed aging infrastructure capable of implementing IoT applications. The concept of cloud computing is to bring cloud computing closer to IoT devices that generate and use data, which in turn brings processing and storage resources closer to users. Rather than moving all processing operations to the cloud, fog computing aims to handle part of the workload produced by applications on devices close to the network to users, called fog servers or fog nodes. These devices can be deployed anywhere from network connectivity: factories, commercial centers, on electric poles, next to railways, inside vehicles, etc. Any device with computing, storage, and network capabilities can be considered a fog node: controllers, switches, routers, embedded servers, surveillance cameras, etc. Placing resources at the edge of the network reduces the time it takes for data to a processing station can be reached in a shorter time, so applications can be optimized for transmission time. However, the processing capability of a fog

node is restricted, thus processing requests with small tasks or small delays will be preferred to process over the fog computing infrastructure, while delay-tolerant and large-scale tasks will still be pushed into cloud layers. Ultimately, fog computing complements the cloud to create a new computing paradigm, cloud-fog computing.

4.0 Literature Review

An extensive literature survey has been done to understand the resource allocation problems, the existing methods that have been applied to solve RAPs in a better and efficient way. It is found in past studies that the exact methods such as Genetic Algorithm, PSO, ACO, and BCO have a size limit and fail to deal when the dimension and the complexity of the problem increases. The findings are discussed below:

Internet business (e-business) is becoming one of the best business models today. To meet the needs of Internet-connected businesses, computing becomes a model that includes services implemented and delivered, such as traditional utilities such as water, electricity, and gas. Many computing paradigms promise to provide such service computing without regard to where is the host or how they provide a service [12].

Frontend interfaces include client devices such as thin clients, sophisticated clients, and mobile devices. Clients need specific interfaces and applications to access cloud computing systems. The server part consists of various servers and data storage systems. A central server is used to manage the cloud system. A central server monitors overall traffic and responds to user requirements in real-time. The main goal of a cloud computing environment is to make the best use of available computing resources. In the optimization process, Scheduling algorithms play an important role. Therefore, user tasks should be scheduled using an efficient scheduling algorithm. Scheduling algorithms are typically aimed at maximizing utilization by distributing the load across the available processors and minimizing the overall execution time [1].

Task scheduling (TS) is one of the most famous combinatorial NP-complete problems [13]. The main goal of scheduling is to schedule tasks in the correct order that can be performed given certain constraints [14].

One of the greatest technology revolutions is undoubtedly the Internet of Things (IoT). Due to the

potential of IoT, daily objects can be programmable (called smart devices) and consciously work in harmony with optimized performances [15] [16]. Therefore, IoT is changing the way how daily objects (such as phones to smartphones, TV to smart TV, etc.) work in our life. Regularly, the number of IoT-capable devices is increasing. The number of smartphones and tablets for 2013 passed one billion and the expected number for 2017 is almost two billion. However, the expected number of IoT device units in 2020 is 20 billion [17].

As the number of Internet-connected devices increases, today's technology is not ready to fully bring the power of IoT in our daily life because a huge amount of data, which is generated by many Internets connected devices, is required to be analyzed. On the other hand, the powerful data management of cloud computing allows IoT to make the revolution in our life. The simplicity of usage, the flexibility of data access, ease of maintenance, time, and energy efficiency, and pay-as-you-go policy have already increased the usage of cloud computing over traditional computing [18] [19].

Fog computing [20], first proposed by Cisco, represents a new model of Cloud computing, which can transform a network edge into a distributed computing infrastructure capable of implementing IoT applications.

Zhao, Liu, Zhang, Hu, and Xi [3] presented GA-based optimizations for segmentation and offline task scheduling, with different computing and storage requirements for heterogeneous systems with heterogeneous computing and communication resources. Adapts to present a computing algorithm.

Ge and Wei [21] have developed a new scheduler that uses GA to make optimization and scheduling decisions by evaluating an entire group of jobs in a job queue.

Guo-Ning, TingLei, and Shuai [22] developed a simulated GA annealing-based algorithm that considers the QoS requirements of different types of consumption operations, which correspond to the operation characteristics of cloud environments.

Zhu, Song, Liu, Gao, and Cheng [23] demonstrated the benefits of multiagent genetic algorithm (MAGA) over traditional GA by developing a load balancing model based on virtualized resource management.

Xiaoli and Yuping Wang [24] proposed an improved two-step multi-objective evolutionary algorithm that accelerates convergence using a modified operator and local search method. This algorithm generated multiple scheduling plans and improved server efficiency through data placement policies and job scheduling strategies.

Chang-Tian and Jiong [25] considered power consumption, performance criteria, and user QoS as targets, and used Dynamic Voltage Scaling (DVS) for two types of ETU_GA (Time Unify Genetic Algorithm) power consumption and Time Double Fitness Genetic power consumption. We have proposed several algorithms. Algorithm (ETDF_GA) - Balances performance and energy.

Jang, Kim, Kim and Lee [26] demonstrated a job scheduling model in which the scheduler calls a GA scheduler function on each scheduling ground and generates a series of schedules ranked according to user satisfaction and virtual machine (VM) availability.

Junwei and Yongsheng [27] presented an improved GA considering the average task completion time, total task completion time, and cost constraints.

Liu, Luo, Zhang, Zhang, and Li [28] have proposed a multi-objective genetic algorithm (MOGA) scheduling algorithm, considering the revenue and energy consumption of access providers. This allows you to dynamically select the appropriate scheduling based on user's real time needs.

Pop, Cristea, Bessis, and Sotiriadis [29] used a reputation-based genetic planning algorithm for independent work to increase benifits and minimize costs.

Pandey, Wu, Guru, and Buyya [30] used PSO-based heuristics to schedule workflows in a cloud environment, consider not only the cost of execution, but also the cost of transferring dependent data.

Netjinda, Sirinaovakul, and Achalakul [31] used the PSO method to transform real particle data into an integral representation of the result to optimize cost, show potential performance in terms of total cost and convergence, and provide various alternatives, get information about changes in user behavior.

Guo, Shao, and Zhao [32] formulate a multi-criteria work allocation model and described the PSO algorithm in the cloud to optimize time and cost.

Guo, Zhao, Shen, and Jiang [33] developed a model that uses the PSO algorithm based on small

position rules, which is faster and saves processing time.

Chiang, Lee, Lee, and Chou [34] proposed an algorithm adopted ACO-TMS with new state transition rules that reduce the time needed to find satisfactory planning results, along with local search procedures that help improve planning results.

Li, Xu, Zhao, Dong, and Wang [35] proposed based on the Load Balanced Ant Colony Optimization (LBACO) algorithm, the cloud job scheduling policy aims to stabilize the overall workload along with optimizing the scope of the worksheet configuration. It also applies to dynamic cloud systems and provides better load balancing on the system.

Our department's research group which is involved in research in Nature Inspired Swarm Intelligence Techniques has been working on Swarm Intelligence techniques and implementing them on problems such as Multicast routing, Multi-Processor/Job-shop scheduling, graph coloring, Knapsack problem, Timetabling problem, etc [37] [38] [39] [40]. This group has so far worked successfully on serial and parallel implementations of single depot multiple objective VRPs [41] [42] [43], where SI techniques such as ACO, PSO, and IWD were used. This gives away and motivation to extend that work onto the serial and parallel implementation of SI techniques to the Multiple Depot VRPs.

5.0 Motivation

Since Resource Allocation problem (RAP) is an NP-hard problem, which means no exact and efficient method can give the optimal solution to the problem in polynomial time. Resource allocation problem has many additional objectives to be optimized and several constraints to be considered, leads to different versions of RAPs that have been introduced in the last few decades. Various approximation methods have already proposed to solve the RAP by several researchers for decades. But several methods have limitations when dimensions (i.e., task execution cost) and complexity (i.e., task execution time, task transferring time) of the problem increases. To overcome these problems, researchers proposed Swarm Optimization (SO) based techniques to solve RAP, which gives near optimal solutions to the problem in comparison to other existing methods. Since then SO techniques emerged as successful approaches to solve large and complex optimization problem and also become an important research area to work on. Therefore, motivated by both i.e.

RAP's practical importance and considerable difficulty and the advantageous features of SO techniques to solve the complex optimization problems, the proposed research problem has been framed. In particular the motivation is to implement the SO techniques to solve certain variants of the Resource Allocation problem in an efficient manner.

Various IoT applications are complex in nature where the solution space is very large. Therefore, using some metaheuristic solutions are beneficial. Swarm optimization is a promising metaheuristic method that therefore can be explored.

Example:

- A parallel PSO (PPSO) algorithm based on multiple objectives was therefore developed to improve the SLA negotiation process between consumers and providers. The main insight of this algorithm is that SLA negotiation can be automated and the PSO can be parallelized to minimize negotiation time and to maximize system throughput, thus increasing the profits of providers.
- A multi-objective PSO (MOPSO) algorithm based on a modified ranking strategy was developed to improve the task scheduling problem in each data center. The novelty of this algorithm lies in using a modified ranking strategy to minimize evaluation time and improve the quality of the results. The algorithm was executed within the constraints of the tight deadline to improve performance in terms of both waiting time and completion time.
- VM allocation was improved by applying a multi-objective PSO to allocate VMs in physical machines after clustering the hosts. Here the novelty lies in applying PSO and K-means when clustering hosts to improve VM allocation and migration, thus maximizing resource utilization and performance whilst reducing power consumption.

6.0 Proposed Research Objectives

It is proposed to:

- To develop new algorithms for analyzing the performance of task scheduling in IoT cloud environment taking single Resource Allocation variants in consideration.
- To explore the possibility of hybridizing swarm optimization and IoT Applications.

- To develop an algorithm to solve Resource Allocation Problem (RAP) in IoT cloud environment.
- To develop a multi-objective model for optimizing task scheduling algorithm using parallel computing, to reduce transferring time, task execution time, and task execution cost.

7.0 Timeline

Table 1: Graphical Representation of Timeline

Task	**1-6 months**	**7-12 months**	**13-18 months**	**19-24 months**	**25-30 months**	**31-36 months**
Literature review & synopsis						
Data collection & analysis						
Experimental observation for objectives 1 & 2						
Experimental observation for objectives 3 & 4						
Results & inferences						
Thesis writing & Submission						

8.0 Platform

PYTHON will be used as the main tool to run simulations, and to compute and analyze the results of the newly developed algorithms

9.0 References

[1] Zomaya, Albert Y., and Michael Wright. "Observations on using genetic-algorithms for channel allocation in mobile computing." *IEEE Transactions on Parallel and Distributed Systems* 13.9 (2002): 948-962.

[2] Lin, C. T. "Comparative based analysis of scheduling algorithms for resource management in cloud computing environment." *JCSE International Journal of Computer Science and Engineering* 1.1 (2013): 17-23.

[3] Lin, C. T. "Comparative based analysis of scheduling algorithms for resource management in cloud computing environment." *JCSE International Journal of Computer Science and Engineering* 1.1 (2013): 17-23.

[4] Zhao, Chenhong, et al. "Independent tasks scheduling based on genetic algorithm in cloud computing." *2009 5th international conference on wireless communications, networking and*

mobile computing. IEEE, 2009.

[5] Huang, Jiwei, Songyuan Li, and Ying Chen. "Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing." *Peer-to-Peer Networking and Applications* 13.5 (2020): 1776-1787.

[6] Richins, Daniel, et al. "AI tax: The hidden cost of AI data center applications." *ACM Transactions on Computer Systems (TOCS)* 37.1-4 (2021): 1-32.

[7] Patel, Sandeep Kumar, and Avtar Singh. "Task Scheduling in Cloud Computing Using Hybrid Meta-Heuristic: A Review." *Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences*. Springer, Singapore, 2021.

[8] Rawat, Pradeep Singh, Robin Singh Bhadoria, and Punit Gupta. "Comparative Study of the Virtual Machine Scheduling Approaches in Cloud Environment." *Smart Systems: Innovations in Computing*. Springer, Singapore, 2022. 411-418.

[9] Lotfi, Hossein. "Multi-Objective Network Reconfiguration and Allocation of Capacitor Units in Radial Distribution System Using an Enhanced Artificial Bee Colony Optimization." *Electric Power Components and Systems* (2022): 1-13.

[10] Elkordy, Ahmed Roushdy, et al. "Compressed coded distributed computing." *IEEE Transactions on Communications* 69.5 (2021): 2773-2783.

[11] Singh, Jaswinder, and Gaurav Dhiman. "A survey on cloud computing approaches." *Materials Today: Proceedings* (2021).

[12] He, Jiawei, et al. "The Application of Big Data in Cyberspace: A Survey." *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE, 2019.

[13] Bhowmik, Trishna, Sanjoy Goswami, and Banani Das. "A Review on Load Balancing Methods for Cloud Computing Based on Ant Colony Optimization, Honey-Bee and Genetic Algorithm." *Data Engineering and Intelligent Computing* (2021): 317-330.

[14] Vijarania, Meenu, Akshat Agrawal, and Mayank Mohan Sharma. "Task Scheduling and Load Balancing Techniques Using Genetic Algorithm in Cloud Computing." *Soft Computing: Theories and Applications*. Springer, Singapore, 2021. 97-105.

[15] Shrivas, Simran, Sonika Shrivastava, and Lalit Purohit. "A Hybrid Approach Using ACO-GA for Task Scheduling in Cloud." *Smart Computing Techniques and Applications*. Springer, Singapore, 2021. 209-217.

[16] Xu, Gangyan, et al. "Cloud asset-enabled integrated IoT platform for lean prefabricated construction." *Automation in Construction* 93 (2018): 123-134.

[17] Janakiraman, Sengathir. "A hybrid ant colony and artificial bee colony optimization algorithm-based cluster head selection for IoT." *Procedia computer science* 143 (2018): 360-366.

[18] W. Kim, "Cloud computing: Today and tomorrow," *Journal of Object Technology,*, vol. 8, pp. 65-72, 2017.

[19] J. Lin, J. Wu, P. Fournier-Viger, Y. Djenouri, C. Chen and Y. Zhang, "A Sanitization Approach to Secure Shared Data in an IoT Environment.," *IEEE Access* , vol. 7, p. 25359–25368, 2019.

[20] C. White and A. o. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing- overview., Fog Computing and the Internet of Things: Extend the Cloud to Where the Things, 2015.

[21] Tanha, Mozhdeh, Mirsaeid Hosseini Shirvani, and Amir Masoud Rahmani. "A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments." *Neural Computing and Applications* 33.24 (2021): 16951-16984.

[22] Gabi, Danlami, et al. "Hybrid cat swarm optimization and simulated annealing for dynamic task scheduling on cloud computing environment." *Journal of Information and Communication Technology* 17.3 (2018): 435-467.

[23] Vasudevan, Meera, et al. "Energy-efficient application assignment in profile-based data center management through a Repairing Genetic Algorithm." *Applied Soft Computing* 67 (2018): 399-408.

[24] Qu, Zhiguo, et al. "A survey of QoS optimization and energy saving in cloud, edge and IoT." *Cloud Computing, Smart Grid and Innovative Frontiers in Telecommunications*. Springer, Cham, 2019. 33-46.

[25] Hazra, Debojyoti, et al. "Energy aware task scheduling algorithms in cloud environment: A

survey." *Smart Computing and Informatics*. Springer, Singapore, 2018. 631-639.

[26] J. Tan and S. Koo, ""“A survey of technologies in the internet of things,”," in *In IEEE International Conference on Distributed Computing in Sensor Systems*, 2014.

[27] Vijarania, Meenu, Akshat Agrawal, and Mayank Mohan Sharma. "Task Scheduling and Load Balancing Techniques Using Genetic Algorithm in Cloud Computing." *Soft Computing: Theories and Applications*. Springer, Singapore, 2021. 97-105.

[28] Abazari, Farzaneh, et al. "MOWS: multi-objective workflow scheduling in cloud computing based on heuristic algorithm." *Simulation Modelling Practice and Theory* 93 (2019): 119-132.

[29] F. C. V. B. N. a. S. S. Pop, "Reputation guided Genetic Scheduling Algorithm for Independent Tasks in Inter-Clouds Environments.," in *In Proceedings of 27th IEEE International Conference on Advanced Information Networking and Applications workshop*, 2016.

[30] W. a. H.-C. Z. Qing, "Optimization of Task Allocation and Knowledge Workers Scheduling Based on Particle Swarm Optimization.," in *In Proceedings of IEEE International Conference on Electric Information and Control Engineering*, 2017.

[31] N. S. B. a. A. T. Netjinda, "Cost Optimization in Cloud Provisioning using Particle Swarm Optimization.," in *In Proceedings of 9th IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information technology*, 2012.

[32] L. S. G. a. Z. S. Guo, "Multi-objective Task Assignment in Cloud Computing by Particle Swarm Optimization.," in *In Proceedings of 8th International Conference on Wireless Communications, Networking and Mobile Computing*, 2012.

[33] Abd Elaziz, Mohamed, and Ibrahim Attiya. "An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing." *Artificial Intelligence Review* 54.5 (2021): 3599-3637.

[34] Rawat, Pradeep Singh, Robin Singh Bhadoria, and Punit Gupta. "Comparative Study of the Virtual Machine Scheduling Approaches in Cloud Environment." *Smart Systems: Innovations in Computing*. Springer, Singapore, 2022. 411-418.

[35] K. X. G. Z. G. D. Y. a. W. D. Li, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization.," in *In Proceedings of Sixth IEEE Annual China Grid Conference*, 2016.

[36] T. M. S. L. R. Mizan, "Modified Bees Life Algorithm for Job Scheduling in Hybrid Cloud," *International Journal of Engineering and Technology* , vol. 2, no. 6, pp. 974-979, 2017.

[37] S. Singh, "Optimization of Complex Mathematical Function using a Novel implementation of Intelligent Water Drops”.," in *Proceedings of International Conference on Soft Computing for Problem Solving (SocPros-2012)*, 2012.

[38] S. Toofani, "The solution to Multicast Routing Using Ant Colony Optimization and Particle Swarm Optimization Technique.," in *SIAI-2012, Two Day UGC Sponsored Seminar on Social Implication of Artificial Intelligence, KMV Jalandhar*, 2012.

[39] S. Afaq, "Multiobjective Optimization Using Swarm Intelligence Techniques: A Review," in *34th National Systems Conference. Surathkal, Mangalore.*, 2010.

[40] S. Afaq, "On the Solutions to the Travelling Salesman Problem using Nature Inspired Computing Techniques.," *IJCSI International Journal of Computer Science,* vol. 4, no. 8, pp. 326-334, 2011.

[41] S. Gupta, "On Solutions to Vehicle Routing Problems using Swarm Intelligence Techniques: A Review.," in *In Conference Proceedings of* International *Conference on Computer, Communication & Computational Sciences (IC4S'16), Ajmer.*, 2016.

[42] S. Afaq, "Swarm intelligence-based soft computing techniques for the solutions to multiobjective optimization problems.," *International journal of computer science issues (IJCSI,* vol. 3, no. 8, pp. 498-510, 2011.

[43] S. Afaq, "A Novel Approach to solve Graph-based TSP using Particle Swarm Optimization Technique.," in *Proceedings of IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).*, 2012.